

**U.S. PATENT APPLICATION**  
**for**  
**SYSTEM AND METHOD FOR VERIFYING LOAN DATA AT DELIVERY**

Inventors: John L. Kemper  
David A. Coleman

## **SYSTEM AND METHOD FOR VERIFYING LOAN DATA AT DELIVERY**

### **PRIORITY**

**[0001]** This application claims the benefit of U.S. Provisional Application 60/436,977, filed December 30, 2002, hereby incorporated by reference in its entirety.

### **FIELD OF THE INVENTION**

**[0002]** This invention relates generally to computer systems and methods used to process data pertaining to financial assets, such as loans, securities, etc. and more particularly to verifying loan data delivered to a secondary mortgage market purchaser.

### **DESCRIPTION OF RELATED ART**

**[0003]** The purchase of a home is typically the largest investment that a person makes. Because of the amount of money required to purchase a home, most home buyers do not have sufficient assets to purchase a home outright on a cash basis. In addition, buyers who have already purchased a home may wish to refinance their home. Therefore, potential homebuyers consult lenders such as banks, credit unions, mortgage companies, savings and loan institutions, state and local housing finance agencies, and so on, to obtain the funds necessary to purchase or refinance their homes. These lenders offer mortgage products to potential home buyers. The lenders who make (originate and fund) mortgage loans directly to home buyers comprise the "primary mortgage market."

**[0004]** When a mortgage is made in the primary mortgage market, the lender can: (i) hold the loan as an investment in its portfolio; or (ii) sell the loan to

investors in the "secondary mortgage market" (e.g., pension funds, insurance companies, securities dealers, financial institutions and various other investors) to replenish its supply of funds. The loan may be sold alone, or in packages of other similar loans, for cash or in exchange for mortgage backed securities which provide lenders with a liquid asset to hold or sell to the secondary market. By choosing to sell its mortgage loans to the secondary mortgage market for cash, or by selling the mortgage backed securities, lenders get a new supply of funds to make more home mortgage loans, thereby assuring home buyers a continual supply of mortgage credit.

[0005] Loans originated by a lender (or alternatively a broker) are typically underwritten before being closed or prior to delivery (i.e. sale) to a purchaser in the secondary mortgage market. Underwriting provides a recommendation whether the loan meets the credit risk and eligibility requirements of a lender for the purposes of its portfolio or a secondary mortgage market purchaser based on a set of loan information provided by the lender. Often, however, the set of loan information may change between underwriting and closing and delivery of the loan (e.g., through continued negotiations between the borrower and lender). The changes to the set of loan information may affect the underwriting decision, the decision whether to purchase the loan, as well as the price for the sale of the loan. Typically, however, a purchaser is only able to determine differences between the set of loan information used for underwriting and the set of loan information for the delivered loan after the loan has been delivered and the sale transaction completed. If there are any yield adjustments due to differences in the set of loan information, the purchaser is faced with requesting such yield adjustments from the seller after the sale is complete, often after a significant lapse of time.

[0006] Therefore, a need exists for a system and method that provides access to underwriting data at delivery. In addition, a need exists for a system and method that facilitates the comparison of underwriting data for a loan and

delivery data for the loan at the time of delivery, before the sale transaction is complete.

**[0007]** In addition, while the sale of loans to the secondary mortgage market in exchange for cash or MBS has worked to increase home ownership rates, these rates could further be improved if new types of investment instruments or assets and new types of mortgage products could be provided. Efforts to offer new types of investment instruments and new types of loan products have been hampered by the fact that current data processing systems for processing loan information (including information on both the borrower side and on the investor side of the process) are not sufficiently efficient and flexible. Modifying the data processing system to support a new type of loan product or a new type of investment instrument is very difficult and expensive. In many cases, inherent limitations in the architecture of such data processing systems make certain types of new loan products or new investment instruments impossible to offer as a practical matter.

#### SUMMARY OF THE INVENTION

**[0008]** In accordance with one aspect of the invention, a method for verifying loan data for a loan being delivered by a seller to a purchaser where the loan has a first set of loan data associated with an underwriting process includes receiving a second set of loan data associated with a delivery process from the seller, retrieving the first set of loan data, and comparing the first set of loan data and the second set of loan data to determine any differences.

**[0009]** In accordance with another aspect of the invention, a system for verifying loan data for a loan being delivered by a seller to a purchaser where the loan has a first set of loan data associated with an underwriting process includes means for receiving a second set of loan data associated with a delivery process from the seller, means for retrieving the first set of loan data.

and means for comparing the first set of loan data and the second set of loan data to determine any differences.

**[0010]** In accordance with yet another aspect of the invention, a method for generating a price for a loan being delivered to a purchaser where the loan has a set of underwriting data having an identifier includes receiving a set of delivery data from a seller, comparing the set of underwriting data to the set of delivery data to identify any differences, determining a price for the loan based on at least one of the delivery data and the underwriting data and upon identifying at least one difference between the set of underwriting data and the set of delivery data, determining a yield adjustment based upon the at least one difference.

**[0011]** In accordance with another aspect of the invention, a system for generating a price for a loan being delivered to a purchaser where the loan has a set of underwriting data having an identifier includes means for receiving a set of delivery data from a seller, means for comparing the set of underwriting data to the set of delivery data to identify any differences, means for determining a price for the loan based on at least one of the delivery data and the underwriting data and means for determining a yield adjustment based upon at least one difference identified between the set of underwriting data and the set of delivery data.

**[0012]** In accordance with an aspect of the invention, a method for verifying loan data for a loan being delivered by a seller to a purchaser where the loan has a set of underwriting data generated by underwriting logic includes receiving a set of delivery data from the seller using delivery logic, accessing the set of underwriting data using an identifier, and comparing the set of underwriting data to the set of delivery data to determine any differences.

**[0013]** In accordance with yet another aspect of the invention, a system for verifying loan data for a loan being delivered by a seller to a purchaser where the loan has a set of underwriting data generated by underwriting logic includes

delivery logic, coupled to the underwriting logic, for receiving a set of delivery data from a seller and processing the set of delivery data. The delivery logic includes comparison logic for comparing the set of underwriting data to the set of delivery data to determine any differences.

**[0014]** In accordance with another aspect of the invention, a method for verifying loan data for a loan being delivered by a loan originator to a lender where the loan having a first set of loan data associated with a underwriting process includes receiving a second set of loan data associated with a loan origination process for the loan from the loan originator, retrieving the first set of loan data and comparing the first set of loan data and the second set of loan data to determine any differences.

**[0015]** Other features and advantages of the present invention will become apparent to those skilled in the art from the following detailed description and accompanying drawings. It should be understood, however, that the detailed description and specific examples, while indicating preferred embodiments of the present invention, are given by way of illustration and not limitation. Many modifications and changes within the scope of the present invention may be made without departing from the spirit thereof, and the invention includes all such modifications.

#### BRIEF DESCRIPTION OF THE DRAWINGS

**[0016]** Figure 1 is a block diagram of a data processing system according to one preferred embodiment.

**[0017]** Figure 2 is a block diagram showing user services logic of the system of Figure 1 in greater detail.

**[0018]** Figures 3A and 3B are block diagrams showing underwriting logic, acquisition logic, servicer and investor reporting logic, and securitization logic of the system of Figure 1 in greater detail.

**[0019]** Figure 4 is a block diagram of the underwriting logic and delivery logic of the system shown Figures 1 and 3A.

**[0020]** Figure 5 illustrates a comparison process for the comparison logic of Figure 4.

**0021]** Figure 6 is a block diagram showing common services logic of Figure 1 in greater detail.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

**[0022]** Referring now to Fig. 1, a computer system 10 for processing data pertaining to financial assets is shown. As shown in Figure 1, the system 10 comprises a data processing system 12, user systems 14, bulk data systems 16, and other data interfaces 18. The data processing system 12 further comprises user services logic 22, a transaction exchange processor 24, underwriting logic 26, acquisition logic 28, servicer and investor reporting logic 30, securitization logic 32, common services logic 34, a data storage system 38, and other data interfaces 18. Herein, although the term "logic" is used in connection with some blocks and the term "processor" is used in connection with other blocks, these two terms are used interchangeably. The term "processor" is used in the generic sense and is not meant to imply a separate discrete unit of processing hardware.

**[0023]** The data processing system 12 is configured for processing data pertaining to financial assets, such as loans and securities. In one embodiment, the data processing system 12 is configured to be used by a participant in the secondary mortgage market. Herein, for convenience, the participant is referred to as a "purchaser," although it should be understood that the purchaser may

participate in the secondary market in other, different, or additional ways (e.g., as a loan guarantor, as a loan securitizer, and so on).

**[0024]** The data processing system 12 is preferably useable to support various types of transactions which may be executed by such a purchaser in connection with one or more loans. For example, the purchaser may purchase loans from lenders or other loan originators as part of a cash execution. The purchased loans may, for example, be held as investments in the purchaser's investment portfolio. Alternatively, the purchaser may create mortgage backed securities (MBS) as part of an MBS execution, or create other financial instruments or assets that are collateralized by cash flows associated with individual loans, including both loans that have been purchased by the purchaser and other loans that have not been purchased by the purchaser. For example, in the case of MBS, the purchaser may acquire a pool of loans, securitize the pool of loans to create MBS that is then sold to investors, and hold the pool of loans in trust for the benefit of the investors. The purchaser may also receive a fee for guaranteeing to holders of MBS or other financial instruments the repayment of the loans by borrowers. The purchaser may also use loans to create other types of financial assets or instruments, for example, by purchasing loans and selling the financial instruments to investors, or by performing such services for other owners of loan assets.

**[0025]** The acquisition logic 28 is preferably useable to perform such operations as receiving information such as loan term, interest rate, principal owed, and other parameters regarding loans when loans are first purchased or otherwise acquired and entered into the data processing system 12. In the case of cash executions, the acquisition logic 28 is also used to perform such operations as receiving commitments for the purchased loans.

**[0026]** The servicer and investor reporting logic 30 is used to process periodic loan data for loan accounting purposes and generate accounting output in



connection with the purchased loans. Herein, the terms "reporting logic" and "servicer and investor reporting logic" are used interchangeably and both refer to logic that is configured to perform loan accounting and generate accounting output (e.g., for purposes of investor reporting, for purposes of managing a loan portfolio, and so on) in connection with a plurality of loans. The servicer and investor reporting logic 30 preferably performs such functions as receiving loan payment data on an ongoing basis from third party servicers. In this regard, it may be noted that the servicer and investor reporting logic 30 in the illustrated embodiment is not used for servicing loans directly but rather interfaces with a third party servicer. Of course, the servicer and investor reporting logic 30 could also be configured to include additional logic for servicing loans, either as part of the servicer and investor reporting logic 30 or as part of another functional block. The accounting output generated by the servicer and investor reporting logic 30 may include such things as accounting, tax, performance/valuation, and/or other relevant financial information for the loans retained in the portfolio or sold, whole or in part.

**[0027]** The securitization logic 32 is used to generate financial assets. Herein, the terms "financial asset generation logic" and "securitization logic" are used interchangeably and refer to any logic that is used to generate/create financial assets. Herein, the term "financial asset" is used generically to refer to any asset that is backed by one or more cash flows, and includes such things as assets that are created entirely for internal data tracking purposes (e.g., in the case of packets which do not represent securities), as well as assets that have external significance (e.g., in the case of MBS or other security). The securitization logic 32 may be used to generate financial assets such as MBS or assets that are tracked internally in situations where the owner/operator of the data processing system 12 purchases a pool of loans and holds the loans as an investment in its own portfolio.

**[0028]** It will be appreciated that the data processing system 12 may perform fewer or additional functions as compared to those described herein. For example, an entity that performs only some of the above-mentioned processes may use a computer system that contains only a subset of the functions described herein. Herein, it will be assumed that the data processing system 12 is used to support each of the business processes described above.

**[0029]** Generally speaking, in the illustrated embodiment, there are three access points for external systems into the data processing system 12. Access can include data flow into and out of system 12. A first access point into the data processing system 12 is the user services logic 22 which provides entry to the user systems 14. A preferred implementation of the user services logic 22 is described in greater detail below in connection with Fig. 2. For purposes of explanation, the user systems 14 are assumed to be operated by human users that participate in some way in the above mentioned business processes. For example, the human user may be an employee of a lender or other loan originator that uploads loan information to the purchaser (or corrects, updates, and so on, information that has previously been provided) in connection with committing to deliver or actually delivering a group of loans to the purchaser, an employee of an owner of a portfolio of loans that uploads loan information in connection with a group of loans the owner wishes to have securitized by the purchaser, an employee of a servicer that uploads payment information regarding a group of loans serviced by the servicer, an employee of an institutional investor that downloads information regarding the financial performance or other data regarding investment instruments created and maintained by the purchaser, an employee of the purchaser itself, and so on.

**[0030]** A second access point into the data processing system 12 is the transaction exchange processor 24 which provides entry to the bulk data systems 16. The transaction exchange processor provides an alternative, bulk transfer mechanism for exchanging at least some of the transaction-related data

mentioned above in connection with the user systems 14, typically without intervention of a human operator. Such bulk data transfers may occur with lenders, servicers, and so on. The transaction exchange processor 24 receives/sends transactions, and prescreens/sorts/translate data if needed, and makes the transactions/data available for further processing in the data processing system 12 or outbound transmission. A third entry access point into the data processing system 12 is through the data interfaces 18. The data interfaces 18 may be used to exchange other types of data between other computer systems and the data processing system 12. For example, the data interfaces 18 may be used to import or export data to other external computer systems (that is, computer systems not under the control of the purchaser) or other internal computer systems (e.g., computer systems that are under the control of the purchaser but that provide functionality that is not integrated into the data processing system 12).

**[0031]** The data processing system 12 is described in greater detail below in connection Figs. 2-6. As will become apparent from the discussion below, the preferred data processing system 12 exhibits a high level of data, service and time granularity. With respect to data granularity, the system 12 is capable of decomposing loans into a series of highly granular cash flows and tracking all of the cash flows from the point the cash flows enter the data processing system 12 (e.g., as part of a loan payment or other cash flow source) to the point the cash flows exit the data processing system 12 (e.g., as part of a payment on a financial instrument). The decomposition of a particular loan into sub-loan cash flows may occur when the loan is first acquired, later when servicing activity begins on the loan, or at another time. When loan payments are received, the allocation of the loan payment into individual cash flows may be performed by logic executed by the servicer, by the data processing system 12, or by other logic. Ideally, all or nearly all of the cash flow sources associated with a particular loan can be identified and tracked. Additionally, it is also possible to

aggregate cash flows from a borrower perspective or other entity perspective. For example, a series of loans (e.g., all to the same borrower) may be aggregated into a higher order cash flow and then the aggregation of the loans may be decomposed. It is also possible to add cash flows to existing loans, for example, so that a new cash flow (e.g., for a new line of credit) may be established without having to set up a new loan. This provides additional flexibility to modify a borrower's loan over time. Thus, the data processing system 12 not only decomposes and maps cash flows associated with such things as principal and borrower paid interest, but also sub-loan level cash flows arising in association with the borrower paid interest or fees associated with the loan such as servicing fees, guarantee fees, mortgage insurance, prepayment penalties, borrower-paid fees, servicer advances, servicer recoveries, and loss/default components, and provides other flexibility. Additional description regarding exemplary possible sources of cash flows is provided at the end of this section. The decomposition and mapping of cash flows dramatically increases the number of different types of financial instruments that may be created, because it makes it possible to create financial instruments based on these other cash flows. In turn, this makes it possible to create financial instruments that are more optimally configured to meet the needs of the owner of the financial instrument.

**[0032]** With respect to service granularity, the data processing system 12 represents loans as a series of attributes and uses a business rules engine to process loan information. This dramatically simplifies the process of expanding the capabilities of the data processing system 12 to process data associated with any type of loan. The capability to process a new type of loan may be added by adding an additional attribute to a list of attributes corresponding to the new product feature (or modifying existing attributes), by using the attribute to indicate the presence or absence (and/or other characteristics) of the new feature in a particular loan, and by modifying the rules engine to incorporate

additional rules regarding the new loan feature. It is not necessary to build a completely new data processing system for the new type of loan. This makes it easier to offer new types of loans which are more optimally configured to meet the needs of individual borrowers. An exemplary set of attributes is described at the end of this section.

**[0033]** With respect to time granularity, the data processing system 12 is capable of processing data using a much smaller time slice or update period than has been possible in the past. In the past, systems have typically been constructed around the assumption that servicers provide monthly reports which summarize loan activity that occurred during the previous month. The time slice for reporting has been one month and sub-monthly temporal data has been lost. In the data processing system 12, when information regarding new loans is received by the acquisition logic 28 and/or when information regarding loan payments is received by the servicer and investor reporting logic 30, this information preferably includes information regarding the date the loan was acquired, the date or dates within each month or other period other period on which a payment or other transaction is expected, and/or the date the payment was received. The time slice in the data processing system 12 is therefore one day (or less, if a smaller time slice such as AM/PM, hour, minutes, seconds, and so on, is used). The temporal information is stored and maintained in databases which are synchronized/commonly accessible by the acquisition logic 28, the servicer and investor reporting logic 30, and the securitization logic 32. As a result, the acquisition logic 28, the servicer and investor reporting logic 30, and the securitization logic 32 each have access to this highly granular temporal information regarding loan acquisitions and payments. The increased time granularity supports the above-mentioned capabilities to offer a wider array of loans to borrowers and a wider array of financial instruments to investors. For example, the increased time granularity facilitates offering loan products in which the borrower is expected to make bi-weekly payments, which may be

attractive to borrowers that get paid bi-weekly instead of twice-monthly or monthly. This also facilitates handling loan products in which the date of a transaction is meaningful, such as daily simple interest loans. Further, because sub-loan cash flows can be processed using a one day time slice (or less), it is possible to create financial instruments based on cash flows that are processed on a per day basis.

**[0034]** Another benefit of the acquisition logic 28, the servicer and investor reporting logic 30, and the securitization logic 32 being provided on a common platform and having access to common/synchronized databases is that each system has an up to date view of the data. The data processing system 12 has the ability to accept payment and other transaction information from a servicer as such transactions occur (e.g., using daily, hourly, or near real-time updates) instead of or in addition to receiving end of the month summary transaction information from the servicer. Once the data is received, it is accessible throughout the data processing system 12. For example, it is not necessary to limit the data updates for the securitization logic to a once-per-month basis at the end of a servicing cycle. Therefore, an up to date view of the data is available throughout the data processing system 12.

**[0035]** It should be apparent that it is also possible to construct data processing systems which do not incorporate the advantages described herein in connection with the data processing system 12, or which also incorporate additional advantages not described herein. Further, it may also be noted that the separation of functionality shown in Figs. 1-6 is necessarily to some extent conceptual, and it is also possible to provide the same functionality in other ways. Additionally, although numerous functions are described below, it may be noted that it may be desirable to provide fewer, additional, or different functions in a given data processing system depending on the application and what is needed.

**[0036]** Referring now to Fig. 2, a preferred implementation of the user services logic 22 and subcomponents thereof will now be described. The user services logic 22 includes electronic registration logic 50, access and security logic 52, user experience logic 54, report request processing logic 62, and a notification processor 64. The registration logic 50 is used to register individual users to be able to use the data processing system 12. For example, an employee of a lender may be given a login name and password to access the data processing system 12. User registration preferably includes providing each user with an authorization profile that defines the extent and type of access the user is given to the data processing system 12 and the types of operations that the user may perform while accessing the data processing system 12. The access and security logic 52 cooperates with the electronic registration logic 50 to permit users to access the data processing system 12 in the manner authorized.

**[0037]** The user experience logic 54 provides a user interface to the data processing system 12. Preferably, the user accesses the data processing system 12 through the Internet or an Intranet by using a personal/laptop computer or other suitable Internet-enabled device. For example, the data processing system 12 may be accessible to users by visiting the purchaser's web site (that is, the web site of the entity that owns/operates the data processing system 12, and that is assumed to be in the business of purchasing, guaranteeing, and/or securitizing loans) and clicking on appropriate links located at the web site. Depending on the authorizations the user has been given in the registration logic 50, the user is able to access different web pages of the web site relating to the underwriting logic 26, the acquisition logic 28, the servicer and investor reporting logic 30, and the securitization logic 32. For example, there may be one or more web pages relating to acquisitions that may be accessed by lenders, one or more pages relating to servicing that may be accessed by servicers, and so on. The user may then perform functions in

accordance with what is permitted by the user's authorization profile (which, in turn, is typically based on the user's employer and the user's job function for that employer). For example, an employee of a lender may be given authorization to access web pages associated with the acquisition logic 28 and commit the lender to deliver a quantity of loans on a future date (i.e., to engage in a forward commitment with the purchaser). The types of operations that different users may perform is described in greater detail in connection with Figs. 3A –3B and 4– 6 below.

**[0038]** The user experience logic 54 includes business application components 56, reference data 58, and user help logic 60. These components provide implementation support to the above-described user interface. The business application components 56 includes logic that assists directing the user to the correct web page. The reference data 58 may include data regarding user preferences for the appearance of web pages to the user. The reference data 58 may also provide general reference data and content that assists user interaction with the web site. The reference data 58 may also include data regarding particular lenders, such as the year the lender was first approved to do business with the purchaser, contact information for the lender, and performance information such as statistics and transfer history for the lender. The user help logic 60 provides other help or "How To" components.

**[0039]** The user services logic 22 also includes report request processing logic 62 and a notification processor 64. The report request processing logic 62 permits lenders and servicers to access the data processing system 12 and request reports generated from the data the lenders or servicers have provided the purchaser. The reports may be predefined "canned" reports, or may be ad hoc reports defined by the user by drilling down into the data and/or defining data filters. The type of reporting generation capability available may be made dependent on the type of user. The report request processing logic 62 may be used for incoming data in connection with lenders and servicers and/or for



outgoing data in connection with investor reporting. Investor reporting may also be handled by other logic described below.

**[0040]** The notification processor 64 sends notifications/alerts to users. For example, the notification processor 64 may be used to send e-mail (or fax, automated telephone call, and so on) to a user associated with a servicer or lender indicating that data which has been submitted by the servicer or lender has been processed, and that the processed data is ready for review. The notification processor 64 is useful in the context of exceptions processing, when lender/servicer data is processed but the processing indicates that there may be an error in the lender's/servicer's data which requires review by a human operator.

**[0041]** Referring now to Fig. 3A, a preferred implementation of the underwriting logic 26 and sub-components thereof will now be described. The underwriting logic 26 is typically accessed by users that originate loans, such as lenders and brokers. The underwriting logic 26 includes data capture logic 70, underwriting logic 74, and credit scoring logic 72. The data capture logic 70 is used to receive information to be used in loan underwriting and appraisal (e.g. information from a loan application and a credit report). Typically, the information that is received for loan underwriting is a subset of the information that would be provided on a loan application.

**[0042]** The credit scoring logic 72 and the underwriting logic 74 cooperate to analyze the information to determine if the loan meets credit risk and eligibility requirements of the purchaser or of a lender for the purposes of its portfolio, and then issue a recommendation based on the assessment of the overall risk profile of the loan. The credit scoring logic 72 generates a credit score of the loan applicant based on the loan applicant's credit history. For example, a credit report for the potential borrower may be obtained once the lender obtains authorization from the potential borrower.

**[0043]** Underwriting logic 74 combines the credit score with other information provided by the potential borrower to the lender to recommend whether to approve the loan. Such information may include, for example, one or more of the following: debt-to income ratios, appraisal value, income, borrower contribution, cash reserves of the borrower, loan-to-value ratio, loan purpose, loan term, loan type, property type, occupancy status and amount of subordinate financing, and other factors. Underwriting logic 74 stores the credit score and the other information and assigns a unique underwriting casefile ID to the lender and loan application for the particular borrower. The recommendation provided by the underwriting logic 74 indicates whether the loan meets the purchaser's credit risk and eligibility requirements. Preferably, the recommendation provides a message such as (1) "approved" or (2) "refer to help center" or like message regarding whether the loan meets the credit risk requirements. Underwriting logic 74 may also provide a message such as (1) "eligible" or (2) "ineligible" regarding whether the loan product meets the purchaser's eligibility requirements. For each loan, the underwriting logic 74 also preferably indicates (i) minimum income and asset verification requirements, (ii) credit related documentation requirements, and (iii) the required level of property field work necessary to complete the processing of the loan file. The underwriting logic 74 may produce the verification and approval requirements for referred loans as well as for loans recommended for approval.

**[0044]** If a loan submitted to the underwriting logic 74 receives a recommendation of "refer", the underwriting logic 74 may also provide the lender with information identifying problem areas with respect to the borrower's application and suggested areas for improving the borrower's chances for approval (e.g., lower loan amount or reduce debt). In addition, the lender may refer the borrower to a help center to receive the benefit of such information and recommendations.

**[0045]** The recommendation provided by the underwriting logic 74 is stored with the associated data provided by the lender and borrower (hereinafter, the credit score, other information and the underwriting recommendation are referred to as "underwriting data"). The underwriting data may be stored in, for example, a database. A lender may resubmit a loan for underwriting (e.g., if property, loan or borrower information changes). Each time a loan is resubmitted to the underwriting logic 26, the underwriting data (including the underwriting recommendation and verification requirements) is updated to include the data from the most recent submission. The information received and generated by underwriting logic 26 is preferably available to the other sub-components of the data processing system 12 as discussed in further detail below with respect to Figures 4 and 5.

**[0046]** The underwriting logic 26 may also be used to generate reports that provide information regarding the underwriting recommendation for a particular loan, information used in determining the recommendation (e.g., property, loan, and borrower information), and information summarizing key statistics from the credit report (e.g., borrower's open accounts, derogatory accounts and undisclosed accounts).

**[0047]** Still referring to Fig. 3A, a preferred implementation of the acquisition logic 28 and sub-components thereof will now be described. The acquisition logic 28 further includes cash committing logic 80, deal management logic 82, lender eligibility logic 84, pricing logic 86, delivery logic 88, certification logic 90, and custody logic 92.

**[0048]** The cash committing logic 80 provides a facility for performing all cash commitment functions. Typically, a master agreement/contract may be in place between the purchaser and the lender which defines overall terms of loan sales to the purchaser pursuant to particular commitments. A cash commitment is an agreement (typically, governed by the overall master agreement) in which the

mortgage purchaser agrees to buy mortgages from mortgage sellers (e.g., lenders) in exchange for a specified price in cash. Typically, a cash commitment agreement specifies the type of mortgage(s) the seller plans to deliver, the amount of time the seller has to make a delivery, the price the mortgage purchaser will pay the seller for the loan(s), other pertinent loan terms, and, in some cases, loan level details pertaining to the mortgage.

**[0049]** The cash committing logic 80 provides a central point for approved lenders (or other approved sellers) and the purchaser to perform all cash commitment functions. These functions may include, for, example, making standard forward commitments, handling pair-off of commitments, extending commitments, over-delivering of a commitment, maintaining configurable parameters, updating contact information, updating commitment records, viewing and selecting from a seller's favorite product list, adding to and maintaining the seller's favorite product list, viewing contracts, fees, prices, yield adjustments, and so on. As previously described, the access and security logic 52 verifies the identity of the user (using a login ID and password) and allows the user to gain access to the cash committing logic 80. Different types of users may be granted different levels of access to the cash commitment logic 80 (e.g., for different employees within a seller organization having different levels of authority to act on behalf of the seller).

**[0050]** In the preferred embodiment, the system 12 includes the ability to limit the different types of loans that a given seller may sell to a subset of the loans which the purchaser may purchase. The different products may comprise loans of different terms, different interest rates and types of interest rates (fixed or variable), as well as a variety of other features or combinations of features that may be offered in connection with the particular mortgage products. This information may be stored in the lender eligibility logic 84, described below, and the cash committing logic 80 may interface with the lender eligibility logic 84 to limit commitment activity to only those products that the seller is eligible to sell.

During the committing process, the seller selects the type of product the seller plans to deliver from a list of eligible products. Sellers may be provided the ability to flag any eligible product as a "favorite," and are able to select products from a favorites list when making commitments. Preferably, sellers are also provided with the option to assign their own marketing name for each eligible product in the seller's favorite list. In another embodiment, rather than selecting from a list of eligible products, sellers may be provided the ability to define a product they plan to deliver by defining the loan attributes.

**[0051]** The committing logic 80 provides sellers with the option to apply a commitment to a master agreement. Information regarding master agreements is supplied by the deal management logic 82 and displayed in the cash committing logic 80 for a given seller. The display may, for example, indicate valid master agreement numbers, the unfulfilled commitment amount in dollars for each master agreement, the expiration date for each master agreement, and/or other pertinent information.

**[0052]** The deal management logic 82 is used to store and track terms of the deals/contracts made between sellers of loans and the purchaser. When a seller contacts the purchaser to initiate negotiation of a new deal, an employee or other representative of the purchaser uses the deal management logic 82 to create a master agreement, MBS pool contract and all the associated variances.

**[0053]** During the master agreement negotiation process, all terms and stipulations of the agreement are entered into the deal management logic 82. The deal management logic 82 enables authorized users creating or modifying variances to identify editable variances and facilitates transforming "codeable" variances into business rules in the delivery logic. The deal management logic 82 also facilitates communication of these variances to users responsible for analyzing them. Users responsible for analyzing variances are provided a link to

the edit engine where they are able to add, modify, or delete edits based on their analyses.

**[0054]** The deal management logic 82 also integrates with the pricing logic 86 so that loan level yield adjustments that reflect negotiated variances may be entered and displayed in the generated master agreement. The seller's specific adjustment tables (referencing master agreement and variance reference numbers) may also be stored in the deal management logic, or, more preferably, in the lender eligibility logic 84.

**[0055]** The lender eligibility logic 84 is logic that maintains information regarding the eligibility of particular lenders to offer particular products made available by the purchaser. The lender eligibility logic 84 allows users (via web interface) to maintain and update product or lender-specific parameters in connection with the committing logic 80, the delivery logic 88, the certification logic 90, the custody logic 92, and the servicer and investor reporting logic 30. The lender eligibility logic 84 may also be used to set pricing incentive adjustments, yield adjustments, fees and other parameters at the lender and product levels.

**[0056]** The pricing logic 86 is the logic used to generate pricing information and provide information to other logic in the data processing system 12, including the underwriting logic 26, the committing logic 80, the delivery logic 88, the certification logic 90, the custody logic 92, and the servicer and investor reporting logic 30. For example, the pricing logic 86 may be accessed during delivery to determine the price to be paid for a particular loan, or after the loan is delivered to determine how changes/corrections in loan information affect pricing. The pricing logic 86 takes into account pricing elements such as commitment/interest price (based on interest and the type of commitment), commitment calculations (e.g., for yield adjustments associated with pair-offs, over delivery, extensions), and credit adjustment price (based on loan level

credit risk). In addition to cash pricing (i.e., pricing in situations where the loan is paid in cash), the pricing logic 86 may also be used for MBS pricing (i.e., pricing in situations where the loan is paid for using a mortgage backed security). The pricing elements related to an MBS includes the guarantee fee, the buy-up/buy-down amount and the credit adjustment amount.

**[0057]** The pricing logic 86 interacts with the delivery logic 88 (described in greater detail below) when a seller is unable to fulfill the terms of its original commitment to generate yield adjustments associated with pair-offs, over delivery, and extensions. The pricing logic 86 acquires delivery and under delivery tolerance amounts from the lender eligibility logic 84, processes data from a commitment inventory database to locate expired commitments and under deliveries, based on input from the delivery logic 88. The pricing logic 86 also processes data associated with the original commitment parameters to generate yield adjustments. Additionally, yield adjustments may also be assessed at the time of delivery for credit risk in connection with one or more loans that exceeds a pre-determined and agreed-upon level. In particular, at the time a cash commitment or MBS deal is made, a certain level of credit risk is assumed when determining the cash price or MBS guarantee fee. Later, when loans are actually delivered, the true risk level is identified. If the cash price or MBS guarantee fee does not account for this actual level of risk, a yield adjustment is made. The system allows the option of selecting either an upfront loan level yield adjustment at the time of delivery or a guarantee fee basis point adjustment to permit the payment to be made over time.

**[0058]** The pricing logic 86 also interacts with the servicer and investor reporting logic 30 when there are loan level changes during the servicing of the loan that result in a request for pricing. The servicing logic 142 sends the pertinent data attributes needed for pricing to the pricing logic 86 and the pricing logic 86 returns pricing information for the loan in question.

**[0059]** The pricing logic 86 may also be used to access prices set forth in pricing grids that store pricing information as a function of various loan parameters and/or features, e.g., interest rate and remaining term in connection with a particular seller. The pricing grids may be generated manually (e.g., in a spreadsheet which is provided to the pricing logic 86) or automatically. The pricing logic 86 may also be used to generate reports regarding pricing information.

**[0060]** The delivery logic 88 is the logic used to process loans when loans are delivered to the purchaser in connection with a purchase. The delivery logic 88 analyzes loan attributes, the associated deal/contract with the seller, and execution parameters to determine if the loan is acceptable for submission under the terms and conditions of the deal. The delivery logic 88 also invokes the pricing logic 86 to determine the price and/or yield adjustment associated with accepting the loan. The delivery logic 88 also allows sellers to set up pools in cases where the loans are pooled in MBS.

**[0061]** The delivery logic 88 receives electronic loan data (hereinafter referred to as "delivery data") by way of the users services logic 22 or the transaction exchange processor 24. The purchaser will generally also receive paper loan documents that support the electronic loan data received by the data processing system 12.

**[0062]** The delivery logic 88 utilizes aspects of the underwriting logic 26, the deal management logic 82, and the pricing logic 86. Each loan that is delivered is checked against business rules and data format rules. Business rules are based on the product, pool/piece/contract, pricing, commitment, and other factors. For example, a seller may inadvertently try to deliver a 15-yr loan in connection with a commitment for 30-yr loans, and the business rules provide a mechanism for identifying that the 15-yr loan can not be used to satisfy that commitment. The delivery logic 88 uses the notification processor 64 to notify



the seller when/if the data that is being delivered does not match the commitment.

**[0063]** The delivery logic 88 is also integrated with the underwriting logic 26. Figure 4 shows a block diagram of the delivery logic and underwriting logic of the system of Figures 1 and 3A. As discussed above, underwriting data (including the underwriting recommendation) for a particular loan is assigned an underwriting casefile ID and may be stored in a database 406. The delivery loan data for a particular loan to be delivered may be assigned a unique loan ID and stored in a database 406. Delivery logic 88 and underwriting logic 26 preferably communicate to share data to perform various functions, such as (i) populate a loan delivery template of the delivery logic 88, (ii) confirm that the loan being delivered meets underwriting criteria, and/or (iii) determine if the delivery data for a loan that was also underwritten using underwriting logic 26 is the same as the underwriting data provided for the loan.

**[0064]** When a seller is entering delivery data for a loan being delivered via delivery logic 88, much of the information required is the same as the information required by underwriting logic 26. As discussed above, the information received for loan underwriting is typically a subset of the information that would be provided on a loan application. If the loan was underwritten using underwriting logic 26, a seller may provide an underwriting casefile ID to the delivery logic 88. Delivery logic 88 may then send a request to the underwriting logic 26 to transfer the current underwriting data associated with the underwriting casefile ID in the database 406 to the delivery logic 88. In an alternative embodiment, the delivery logic 88 may directly access the underwriting data associated with the underwriting casefile ID in the database 406. The underwriting data transferred may include one or more of the following: borrower social security number, co-borrower social security number (if applicable), property address, loan amortization type, loan guarantor type,

number of units, loan processing method, employment information, loan type, underwriting results, loan term, LTV, CLTV, borrower reserves, and so on.

**[0065]** The underwriting data may be used to populate a delivery template presented to a seller delivering the loan by, for example, a web interface. Preferably, the underwriting data is identified (e.g. italicized or shaded) so the seller can identify the data obtained from the underwriting logic 26. The seller, therefore, does not have to reenter loan data previously entered during the underwriting process. In addition, the seller can review the underwriting data to ensure that it is accurate. The seller, preferably, can edit or update the underwriting data, if necessary. For example, the underwriting data may be edited by way of a computer system with a user interface to receive inputs regarding the underwriting data. If a seller does not provide an underwriting casefile ID or if the loan being delivered was not underwritten using underwriting logic 26, the seller will have to enter all of the loan data required for delivery of the loan. The seller may enter the loan data using, for example, a keyboard or other input device in conjunction with a computer system.

**[0066]** If a loan being delivered was not previously underwritten using underwriting logic 26, delivery logic 88 may interact with underwriting logic 26 to confirm the loan being delivered meets the underwriting criteria of the purchaser. Delivery logic 88 may transfer delivery data provided by the seller at delivery to the underwriting logic 26. Underwriting logic 26 will then process the delivery data relevant to providing an underwriting recommendation, as described above, to provide an underwriting recommendation. Underwriting logic 26 then returns underwriting data (including the underwriting recommendation) to the delivery logic 88. The underwriting data and, in particular, the underwriting recommendation, may be provided to the seller using notification processor 64 (Figure 2). In addition, the underwriting data may be used by the pricing engine 86 (Figure 3A) to determine a price for the sale of the loan.

**[0067]** Delivery logic 88 also includes comparison logic 410. Generally, certain terms of the delivery data of a closed loan being delivered to the purchaser (e.g., occupancy type, product type, amortization type, loan term, property type, loan purpose, property sales price, appraised value, etc.) should match the underwriting data provided to the underwriting logic 26. If certain material terms do not match there may be an impact on the underwriting decision and the price and/or yield (e.g., credit related yield adjustments or fees) the seller will be charged. Preferably, material terms are determined using the business rules, i.e., business rules may be created which identify what terms are material. Accordingly, comparison logic 410 permits the comparison of the underwriting data for a particular loan to the delivery data provided for the loan to determine any differences.

**[0068]** In an alternative embodiment, comparison logic 410 may be implemented as a separate system from the delivery logic 88 and the data processing system 12. Accordingly, comparison logic 410 may be configured to compare the underwriting data for a particular loan (as provided by underwriting logic 26 and database 406) to loan data provided by an external data source (i.e., a data source that is not related to the data processing system 12). For example, a loan sale transaction may occur between two lenders, e.g., a loan originator and a warehouse lender. Alternatively, a lender may sell a loan it owns but did not originate. The loan originator may use underwriting logic 26 to underwrite the loan and the warehouse lender (or purchaser) may wish to compare the loan data provided by the loan originator to the underwriting data before purchasing the loan.

**[0069]** Figure 5 illustrates a comparison process for the comparison logic 410. At block 510, the delivery data for the loan being delivered is provided to the delivery logic 88. As mentioned above, delivery logic 88 receives electronic loan data from a seller by way of user services logic 22 (Figure 1) or the transaction processor 24 (Figure 1). If the loan does not have an underwriting

casefile ID (decision 504), the delivery process proceeds as described with respect to Figure 3A. Preferably, if the loan does not have an underwriting casefile ID (i.e., the loan was not underwritten using underwriting logic 26), the loan is submitted to underwriting logic 26, as described above, to determine whether the loan meets the purchasers underwriting criteria.

**[0070]** If the loan has an underwriting casefile ID (decision 504), the delivery logic 88 sends a request to the underwriting logic 26 to retrieve the underwriting data associated with the underwriting casefile ID at block 508. Underwriting logic 26 returns the underwriting data associated with the underwriting casefile ID to the delivery logic 88. As mentioned above, in an alternative embodiment, the delivery logic 88 may directly access the underwriting data associated with the underwriting casefile ID in the database 406. The underwriting data and delivery data are then compared to determine any differences in a predetermined set of information (e.g., loan, borrower, and property information) at block 510. If the predetermined set of loan information of the underwriting data and the delivery data are the same (decision 512), the delivery process proceeds as described herein with respect to Figure 3A at block 514. If the predetermined loan information from the underwriting data and the delivery data are different (decision 512), it is determined whether the differences are material to the underwriting decision or the price and/or any possible credit related adjustments for the loan (decision 516). As discussed above, whether a difference is material is preferably determined using the business rules. If the difference is material (i.e., the difference will impact the underwriting decision and price and/or credit related yield adjustment), delivery logic 88 invokes pricing logic 86 (Figure 3A) to determine the price and/or yield (e.g., any credit related yield adjustments or fees) at block 518. Delivery logic 88 may also invoke underwriting logic 26 to underwrite the loan based on the delivery data provided by the seller.

**[0071]** At block 520, the comparison results may be provided to the seller and/or the seller may be notified using the notification processor 64 (Figure 2) when the predetermined set of information from the delivery data does not match the underwriting data and whether there is an impact on the underwriting decision and the price as well as any additional adjustments (e.g., credit related yield adjustments) that may be charged. Preferably, the seller has the option to edit the delivery data provided (e.g. if the seller made an error entering the delivery data). For example, the underwriting data may be edited by way of a computer system with a user interface to receive inputs regarding the delivery data. If the seller does not edit the delivery data (decision 522), the delivery process proceeds as described herein with respect to Figure 3A at block 524. If the seller does edit the delivery data (decision 522), the process returns to block 510 and the edited delivery data is compared to the underwriting data. Also, the seller may decide not to deliver the loan based on the comparison results or the purchaser may decide not to purchase the loan based on the comparison results.

**[0072]** If the difference between the delivery data and the underwriting data are not material (decision 516), the seller preferably has the option to edit the delivery data (e.g. if the seller made an error entering the delivery data into the system). If the seller does not edit the delivery data (decision 526), the delivery process proceeds as described herein with respect to Figure 3A at block 528. If the seller does edit the delivery data (decision 526), the process returns to block 510 and the edited delivery data is compared to the underwriting data.

**[0073]** In another embodiment, a lender may compare underwriting data for a loan with loan data provided by a broker from a loan origination system (hereinafter referred to as "loan origination data"). Loans may be originated by a broker who, once the loans have closed, provides the loans to a lender for funding. As described above, the lender can then either: (i) hold the loans as investments in its portfolio, or (ii) sell the loans to investors in the "secondary

mortgage market” to replenish its supply of funds. Alternatively, a lender may wish to evaluate its loan portfolio by comparing the loan data for loans already in the portfolio to underwriting data for the loans. Such an evaluation may be performed as part of, for example, a decision to sell loans from the portfolio to the secondary market, a quality control process, due diligence, etc. Other parties with an interest in a transaction involving a loan may also wish to evaluate the loan by comparing the current loan data to underwriting data for the loan.

**[0074]** Underwriting logic 26 may be used by the broker to underwrite a loan as described above. When a loan is submitted to the lender by a broker, including the loan origination data, the lender may wish to determine if the loan origination data (or a predetermined subset of the loan origination data) is the same as the underwriting data (or a predetermined subset of the underwriting data) for the loan before accepting the loan from the broker. It possible that the loan data used to underwrite the loan may change before the loan is provided to the lender (e.g., due to continued negotiations between the broker and the borrower). The underwriting data for a loan, identified by an underwriting casefile ID, may be accessed using, for example, user services logic 22 (Figures 1 and 2). The lender may then compare the underwriting data to the loan origination data provided by the broker. The broker may provide the loan origination data, for example, electronically or using physical documents. If there are differences between the underwriting data for the loan and the loan origination data for the loan, the lender may decide not to accept the loan from the broker.

**[0075]** In yet another embodiment, a purchaser such as a wholesaler or warehouse lender, may compare underwriting data for a loan with loan data provided by another lender (or seller). Loans originated by one lender may be sold to another lender, e.g. a wholesaler or warehouse lender. Alternatively, a lender may also sell loans it owns but did not originate. The purchaser may wish

to determine if the loan data submitted by the lender is the same as the underwriting data for the loan before accepting the loan. The process for comparing the loan data and underwriting data is similar to that described above with respect to the broker/lender embodiment.

**[0076]** Returning to Figure 3A, the delivery logic 88 allows the user to edit the delivery data for format/field edits and standard/custom edits necessary to deliver loans to the purchaser. Users have a real time view of updates to the delivery data in order to resolve data errors before the loan is purchased or securitized. For example, if the data indicates that a 15-yr loan is being used to satisfy a commitment for a 30-yr loan, the user may edit the data to indicate that the loan is a 30-yr loan (in a situation where the loan data was incorrectly entered and what was originally indicated as being a 15-yr loan is in fact a 30-yr loan). Alternatively, the user may edit the data to instead apply the 15-yr loan to a different commitment for a 15-yr loan. As a further alternative, the user may edit the data to substitute a 30-yr loan for the 15-yr loan. The delivery logic 88 also includes logic for address correction (detecting erroneous address information and correcting the address information) and geographic coding (to provide additional geographic information on the property, such as longitude and latitude, tract, congressional district, metropolitan statistical area number, and so on). By the end of the process, the delivery logic also generates a unique loan number for each of the loans for tracking purposes.

**[0077]** The certification logic 90 is logic that supports the process of ensuring that all loan documentation is complete and legally binding and that the paper documentation matches the electronic information delivered by the seller. The certification logic 90 generates, stores and makes available to other aspects of the data processing system 12 information pertaining to which loans have been certified. The certification logic 90 is also able to generate custom reports regarding certification data including reports on loans that have not been certified so that appropriate action may be taken (e.g., having the seller

repurchase the loan). The certification logic 90 facilitates data modification and facilitates data matching when loans are redelivered or resubmitted. The certification logic 90 also generates reports to support management decisions with respect to certification activities.

**[0078]** The custody logic 92 is logic that is used to support the custody process, or the process whereby the purchaser stores the paper loan documents during the time from when the loans are purchased or securitized until they are released. Custody protects the physical evidence of investment in negotiable assets. The custody logic 92 manages custodial profile/contact information, custodian/seller relationships, and seller/servicer profile/eligibility information related to custody activities. The custody logic 92 also permits information to be retrieved regarding loan investors. If the market purchaser performs the custody function itself rather than having a third party act as custodian, the custody logic 92 also supports document management in connection with incoming and outgoing documents. In particular, the custody logic 92 tracks when loan documents are in the possession of the purchaser and otherwise manages and monitors the position of the physical loan documents. The custody logic 92 also manages and calculates fees charged for custodial and certification services.

**[0079]** The acquisition logic 28 may also include other logic in addition to the logic described above. For example, the acquisition logic 28 may further include payable/receivable manager logic to track the billing of yield adjustments and fees generated by transactions in the committing logic 80, the pricing logic 86, the delivery logic 88, the custody logic 92, and certain aspects of the servicer and investor reporting logic 30. The payable/receivable manager logic may also be used to display the status (including payment status) of such yield adjustments and fees in a consolidated manner.



**[0080]** Referring now to Fig. 3B, a preferred implementation of the servicer and investor reporting logic 30 will now be described in greater detail. The servicer and investor reporting logic 30 includes loan process and compare (LPC) logic 100, which monitors and verifies the activities of third party mortgage servicers on an ongoing basis. Alternatively, if servicing is performed internally by the owner of the data processing system 12 and is included as part of the servicer and investor reporting logic 30 or as part of another functional block of the data processing system 12, the LPC logic 100 may be used to verify internally generated reporting information. Thus, the LPC logic 100 performs such operations as receiving and validating reporting information pertaining to loan activity, loan delinquency information and unpaid balance comparison reported by the servicer, updating the records of the data processing system 12 regarding the status of all reported loans, and determining the remittance and disbursement amounts that are expected for the loans.

**[0081]** As an initial matter, prior to loan servicing, a comparison is performed of the servicer's data for loans being serviced with the purchaser's data for the same loans. Even if the purchaser's data has already been compared with lender data for the same group of loans, the servicer's data may for some reason be different. Accordingly, the purchaser may provide a predefined set of acquisition data to the servicer that the servicer can compare with the servicer's data. At any time thereafter, the servicer may perform individual queries of the loan data stored on the purchasers data base via the user services logic 22 (web interface) and download the data for further comparison purposes. When exceptions are noted, the servicer can correct its data or submit a change request via the user interface to the attribute change processor (ACP) logic 122, described below.

**[0082]** During the life of the loan, when loan activity occurs (e.g., when the borrower makes loan payments), the LPC logic 100 is executed with regard to a particular loan when a servicer reports transactions to the purchaser. A loan

activity processor 102 handles expected and scheduled servicing transactions including payments, rate changes, curtailments, and so on. The activity processor 102 receives and validates loan transaction data, such as loan activity, unpaid balance comparison, and delinquency status updates. The activity processor 102 can also be configured to check for duplicate transactions, validate servicer information, determine and validate the type of loan transaction, and validate that the loan activity is being reported in the correct reporting period. The activity processor 102 also confirms that changes in unpaid balance and last paid installment are correct, derives expected interest remittance, derives expected principal remittance, and compares the derived amounts to the reported remittance amounts. After validation, the status of the loan is made available to the servicer through the user services logic 22. The activity processor 102 also triggers the appropriate cash and accounting transactions in a book and tax accounting processor 146. When loan activity is processed and does not match the purchaser's expectations based on rules and calculations, exceptions are noted and communicated to users using the notification processor 64.

**[0083]** The amortization/calculation processor 104 is used by the activity processor 102 to calculate loan level amounts, such as principal and interest due, servicing fees and other data pertinent to each loan. Processor 104 may additionally be used to compute derived or decomposed cash flows, such as a guaranty fee or a servicing fee. Business rules are used to identify scheduled and unscheduled principal, calculate fees, calculate remittance and disbursement amounts, calculate amounts to be disbursed to investors, amortization, and accruals. These calculations are used throughout the system 12 to perform functions such as collecting remittances from servicers, dispersing funds to investors and performing accounting activities. The results of processing are available through an interactive user interface to both personnel of the purchaser

and personnel of the servicer for correction when transactions do not comply with business rules.

**[0084]** The trial balance processor 106 provides for validation of parameters such as servicer number, purchaser and servicers loan numbers, effective date, ending unpaid balance, note rate, pass through rate, principal and interest payment, last paid installment (LPI) date, pool number, accrued interest receivable balance, available line of credit, conversion date, reverse mortgage payment, net principal limit, taxes and insurance set asides, property charges set asides, repairs set asides, servicing fees set asides, scheduled payments, and so on. Any discrepancies are resolved and any system updates (loan attribute changes, data updates) are implemented. The LPC logic 100 then reprocesses the activity based on the corrected data.

**[0085]** In addition to borrower payments, the LPC logic 100 may also be triggered with regard to a particular loan when the attribute change processor (ACP) logic 122 makes a change to attributes that affect loan processing or when a loan attribute triggers processing, such as note rate changes, payment changes and loan reporting. The LPC logic 100 may also be triggered by borrower behavior (e.g., loan delinquencies status) at beginning and end of accounting periods.

**[0086]** The servicing event processor 108 identifies and handles business events that are not identified by the activity processor 102. Examples of these events include identifying delinquent loans and identifying loans that are eligible for reclassification or substitution. The delinquency status reporting processor 110 accepts delinquency reasons from the servicer for loans that have payments that are in arrears.

**[0087]** The attribute change processor (ACP) logic 122 processes loan or security level changes. ACP logic 122 processes attribute changes regarding loans. As previously described, in the preferred embodiments, loans are

characterized in the data processing system 12 by a series of attributes rather than by product codes. Each mortgage product that is purchased is then represented by a series of attributes instead of or in addition to an overall product code. New products may be created by creating new combinations of attributes, or by adding new attributes. An exemplary list of possible attributes that may be used is provided at the end of this section.

**[0088]** ACP logic 122 processes attribute changes that occur after loans are brought into the data processing system 12. In particular, after loans are brought into the data processing system 12, the ACP logic 122 processes attribute changes that are unexpected or are unscheduled whereas the LPC logic 100 handles attribute changes that are both expected and scheduled. The ACP logic 122 also validates the attribute change request, assesses the financial impact of the change, updates the appropriate data and triggers the appropriate cash and accounting transactions.

**[0089]** Unexpected attribute changes are changes that are required due to new features or discrepancies between contract documentation and data captured by the acquisition logic 28, this can include changes to loan data and/or changes in loan behavior. Unscheduled attribute changes are changes that may occur based on contract documentation but the timeframe is unknown. For example, an unexpected attribute change would be a change for a daily simple interest cash loan that the purchaser has purchased without knowledge of a particular feature. After the purchase, the borrower exercises options under the feature and the servicer advances the next due date of the loan and submits a loan activity transaction record to the purchaser. Not knowing about the feature, the purchaser rejects the transaction since the loan record does not indicate the presence of the feature. After assessing the exception and evaluating the change, the servicer submits an attribute change request to add this feature and keep the loan in the purchaser's portfolio or in the security, pending confirmation of continued loan eligibility. An example of an unexpected

and unscheduled attribute change would be the case where the lender submits an adjustable rate mortgage change request for a loan that the purchaser has set up as a fixed rate mortgage. The request is processed as an unscheduled change because the purchaser's systems have never had an event scheduled to trigger the change. An example of an unscheduled change is a fixed rate convertible loan which has the conversion option indicated in the terms of the note. It is anticipated that an attribute change will occur but the timing of the event is unknown and therefore unscheduled. The two primary types of unexpected attributed changes are post purchase adjustments (data corrections) and modifications (attribute changes driven by a number of business requirements, such as product flexibility, delinquency management, and substitutions/reclassifications).

**[0090]** In operation, the ACP logic 122 receives attribute change requests which indicate current database values for the loan and the proposed changes. The validation of the loan with the new values is then accomplished by applying the rules processor 180 (Figure 6) to the ACP transaction. The business rules engine is applied to determine whether the changes are allowable and any failed business rules are provided to an operator for further review. Next, the original terms of the contract are used to determine any pricing adjustments of the attribute change. The system determines the difference between the current or adjusted price as applicable and the new price for the purchase adjustments. Next, a human operator reviews the requested change, the impact of the requested change, and any required hard copy documentation needed to justify the change. The operator/business analyst either approves or rejects the change. Rejected transactions may be modified and resubmitted. Approved adjustment transaction values are applied to the database and an audit trail history is maintained. If the result of the change request has an accounting impact, the ACP logic 122 also generates the appropriate transactions to trigger the accounting processor 146.

**[0091]** The ACP logic 122 also includes loan conversion request processing logic 338 for handling loan conversion requests. Thus, when a loan conversion request is received, this logic tracks the request for the change, determines the allowability of the change based on business rules, and employs the remainder of the ACP logic 122 to make the change.

**[0092]** The securities aggregation and management (SAM) logic 130 receives the loan level cash flow information produced by the LPC logic 100 and aggregates this cash flow information to produce security level information. The security level information is produced at each of the following levels: remittance/express date level within each piece/single pool; single pool level or piece level within each major pool; pseudo pool (pool-like reporting group) level; major header level for each major pool; choice pool level; strip level; mega pool level; and mega in mega (MIM) pool level. In addition to securities, the SAM logic 130 is also capable of processing and managing any grouping of loans, cash flows from loans, and other financial instruments. Using a packet activity processor 132, the SAM logic 130 determines the loans in a given pool, aggregates cash flows based on the pool and loan level attributes for all the loans and then updates the system database. The packet activity processor 132 has the flexibility to aggregate loan level cash flows at the most granular level to security level enabling the SAM logic to also manage specific cash flow strips (e.g., access yield strips, interest only strips). At the end of appropriate processing periods, the SAM logic 130 finalizes the relevant security information. The SAM logic 130 then uses a packet disclosure processor 134 to make final remittance level principal and interest, guaranty fee, and other draft amounts available to a cash processing logic 144 and to make security accounting data available to a book and tax accounting logic 146. The SAM logic 130 also calculates, at the various MBS security levels, disclosure data for investors and the MBS payment distribution to investors. The SAM logic 130 also includes packet modification request processing logic which is used to

modify packets in generally the same manner that the attributes of loans are modified as described above in connection with the ACP logic 122. The operation of the SAM logic 130, and in particular packets and the packet activity processor 132, is described in greater detail in connection with the packeting logic 154.

**[0093]** Further, the SAM logic 130 can be used to facilitate the provision of real-time data updating. for example, investors may be supplied with real-time analytic data. the analytic data may include any data that allows investors to more accurately determine the value of their holdings, such as data concerning monthly loan payments, loan prepayments, loan pay-offs, and so on. For example, when a loan pays off, investors may be provided immediate access to this information rather than waiting until the next MBS reporting cycle.

**[0094]** In the illustrated embodiment, the servicer and investor reporting logic 30 and the securitization logic 32 utilize the same data base (see Fig. 6). As a result, the data used by the securitization logic 32 is always synchronized with the data used by the servicer and investor reporting logic 30. Thus, it is not necessary for the securitization logic 32 to wait until the end of a periodic (e.g., monthly) reporting cycle to receive updated data, but rather the securitization logic 32 always has access to up-to-date loan information. In another embodiment, the servicer and investor reporting logic 30 and the securitization logic 32 may utilize different data bases that are synchronized on a weekly basis, on a daily basis, on a sub-daily basis, or in real time, depending on the frequency of update that is desired.

**[0095]** A servicing transfer logic 142 facilitates the process of transferring loans for the servicing rights of owned or securitized mortgages from one servicer to another or from one portfolio to another within the same servicer as of an effective date. A servicing transfer may be initiated, for example, if a servicer decides to stop servicing loans for business reasons, if a servicer

decides to transfer a certain group of loans to another branch or portfolio, if a servicer is involved in a merger or acquisition of the servicer necessitating a transfer to the surviving entity, or for other reasons. The servicing logic 142 processes information regarding the old and new servicers and the loans that are subject to the change in servicing and updates loan record data for the respective affected loans. The effective date of the change in servicing is also specified. Information that is provided to the servicing transfer logic 142 as part of a servicing request includes the transferors servicer number, address and contact information, the transferees servicer number, address and contact information, unique loan numbers to be transferred, effective date, and other data. Additional steps, such as notifying the transferor of the termination and assessing transfer fees may also be performed.

**[0096]** The cash processor 144 provides a facility to allow servicers and other vendors to create and maintain bank account information. The accounts are bank accounts established with the purchaser to facilitate loan transactions. Servicers have the ability to create/select/update their account information in real time, including account numbers and remittance/disbursement information. The information captured in this process allows the cash processor 144 to create and execute Automated Clearing House (ACH) transactions. Historical records of servicers and vendors account and draft information is maintained to assist in resolving any issues that may arise.

**[0097]** Additionally, the cash processor 144 retrieves remittance and disbursement information from other areas of the data processing system 12. The remittance and disbursement information includes effective date, loan number, dollar amount, remittance code, and granular level details. The cash processor 144 performs a rollup of loan level details by servicer number as required. The cash processor 144 also performs a rollup of loan level details by seller number whenever the seller is not the designated servicer. The cash processor 144 triggers appropriate accounting transaction codes as needed that



allow the book and tax accounting processor 146 to record applicable accounting entries.

**[0098]** Finally, the cash processor 144 creates cash transactions, for example, automated clearing house (ACH) transactions, outgoing check transactions, and so on. The cash processor 144 begins this process after the cash processor 144 has completed the process of assessing and validating remittance and disbursement data. The first step in creating a cash transaction is validating servicer/vendor bank account information. Ultimately, an ACH transaction is created that debits or credits the appropriate custodial bank account

**[0099]** The book and tax accounting logic 146 manages accounting activities associated with the loans. The accounting logic 146 provides a consistent methodology for the recording of accounting events related to mortgage business activities across the acquisition logic 28 and the servicer and investor reporting logic 30 into subsidiary ledgers for posting to a general ledger. The book and tax accounting logic 146 supports the accounting activities related to the packaging of loan cash flows to the first level packet for the securitization logic 32. In addition, the book and tax accounting logic 146 supports the accounting activities related to forming securities or packets out of portfolio loan collateral. The investment accounting for securities held in portfolio and for the payment distribution on mortgage derivatives could also be handled by the book and tax accounting logic 146 or, preferably, is handled by separate accounting logic 156, described in greater detail below

**[0100]** The book and tax accounting logic 146 journalizes mortgage related business activity, maintains subsidiary ledgers, provides audit trails, provides data integrity and control within the subsidiary ledgers, facilitates timely reconciliations, provides flexibility to account for new products or changes depending on actual accounting methodologies, and provides information needed

to perform financial analysis. In one embodiment, the book and tax accounting logic 146 utilizes an accounting matrix which is a two-dimensional structure comprised of accounting "families" and "family members." The families are groups of accounting relevant transaction and loan attributes, and the family members are the allowable values for each of the groups. All intersections of families and family members have a debit and credit account number associated with each of the intersections. When the journal entry is created, the appropriate debit and credit account numbers are first assigned to each of the transactions as they are processed. The accounting matrix uses business rules processor 180 to automatically interpret the transactions. As new products are introduced, the accounting matrix is modified to incorporate new family and/or family members to properly record the new business activity. Similarly, as products become obsolete, or as the requirement for breaking out activity on the corporate general ledger becomes less detailed, the accounting matrix can be modified to adapt to those changes as well.

**[0101]** As business activities are processed, they are recorded/journalized in a subsidiary ledger according to the debit and credit account numbers assigned from the accounting matrix. This occurs by translating business activities into family and family member transactions that can be interpreted by the matrix. A subsidiary ledger provides the capability to view the lowest level of business activity that created the entry in the subsidiary ledger to maintain an audit trail for the subsidiary ledger activity. As activity is recorded, a system walk forward test of the subsidiary ledger balances is also performed to assure data integrity with the subsidiary ledger. At the end of accounting cycles, activity within the subsidiary ledgers is automatically summarized and posted to the general ledger

**[0102]** At the end of the accounting cycle, reconciliation is performed between the subsidiary ledger activity and balances, and the general ledger activity and balances using an automated reconciliation tool. An automated reconciliation tool may be provided that generates the results of the

reconciliation and, through a user interface, displays the results to an operator. Any reconciling items between the subsidiary and general ledgers may be analyzed and resolved by the operator. Through the operator interface, the operator updates the status of the reconciling items to indicate the results of the analysis. As reconciling items are resolved, the operator triggers the automated reconciliation facility to repeat the reconciliation and display the results.

**[0103]** The book and tax accounting logic 146 also provides information for financial and operational analysis. Information related to the status of the book and tax accounting logic is provided to operations through an accounting console. The accounting console is a management and operational workflow tool that includes notifications and status information related to the book and tax accounting processes. It also provides summarized reports and the ability to view the detailed information supporting those reports.

**[0104]** A preferred implementation of the securitization logic 32 and sub-components thereof will now be described. The securitization logic 32 includes sifting/sorting logic 152 which accesses inventory, identifies collateral or asset attributes and sub-attributes, and categorizes data at its most granular level in both aggregating and segregating cashflows associated with mortgage assets. The sifting/sorting logic 152 provides a user interactive application that allows users to define selection criteria (loan and/or atomic characteristics), prioritize them, evaluate results, and make decisions about market transactions and their related economics. By sifting and sorting through available inventories, cashflows may be qualified and quantified for optimal aggregation of targeted transactions, given relative market value. The sifting/sorting logic 152 operates under a user maintainable library of business rules associated with mortgage instruments and respective cashflows. An auto sift function is also provided to allow to batch processing of predefined inventory types. For example, a daily auto sift may be executed against "available for sale" loans to aggregate and pre-packet the loans for future transactions.

**[0105]** The purpose of the sifting/sorting logic 152 is to provide a mechanism by which users can examine the entire collateral universe and pair down to smaller groupings of collateral or assets within the universe. Collateral refers to any cash flow derived from loans, pools, securities, commitments, and packets. The purpose of sorting is to group the subset of collateral identified in the sifting process and organize it by a single or multiple attributes to further refine the pool of candidate collateral to be placed into a potential packet. The sifting/sorting logic 152 supports the packeting logic 154, described below.

**[0106]** The packeting logic 154 is used to create, maintain, and otherwise support packets. A packet is an aggregation or packaging of cash flows that is treated as an entity separate and distinct from the incoming cash flows that support the packet and from the cash flows that result from the packet. Packets maintain the data integrity of the underlying assets as received by the LPC logic 100 and create an information chain that maps to a higher-order asset (e.g., an MBS or other financial instrument to be sold to an investor). The source data for packets may be loan-level or packet-level information, and the packets themselves may represent actual securities or just a unit of reporting and remittance.

**[0107]** Packets permit the data processing system 12 to enable and support new transactions by providing a platform for sourcing, normalizing, and centralizing cash flow-related data and building the linkages between loan assets and securities or non-securitized assets. Packets provide greater flexibility in the transformation of cash flows from the primary mortgage/loan level to the secondary market and within the secondary market. Packets provide the flexibility not only to create and sell securities to investors but also to support non-securitized forms of packaging to enable selling or retaining cash flows from individual loans. The ability to create and manipulate packets enables the creation of new types of financial instruments and new types of transactions within the secondary market.

**[0108]** The accounting logic 156 supports additional accounting functions for the securitization logic 32 that are not already supported by the book and tax accounting processor 146. In general, the book and tax accounting processor 146 is responsible for performing maintenance accounting at the loan level (i.e., posting transactions), while the accounting logic 156 is responsible for the accounting logic associated with transformative accounting events.

Transformative accounting events include, for example, securitization events (in which a loan is to be construed to be sold). Other transformative events include a securitization event in which only a portion of the cash flows are sold, a sale event of a portfolio securities, and a sale event involving a whole loan. In addition, the accounting logic 156 is responsible for ongoing maintenance in connection with the reconciliation of securities cash payables. The accounting logic 156 performs such things as deriving the initial cost basis at the time of acquisition for every loan and inventory, maintaining the cost basis of each loan, tracking accounting intent for each loan, and performing market valuation for each loan. Of course, although the functionality of blocks 146 and 156 are shown as being conceptually separate, this functionality could also be combined.

**[0109]** The position monitor 158 allows monitoring of the purchaser's overall trade and investment position. Particularly, the position monitor 158 is an interactive tool that is usable to monitor positions of investors of whole loans and securities, and designate or redesignate inventory between trading accounts. The position monitor 158 is able to provide this information in near real time because the position monitor 158 either uses the same transactional database(s) as the servicer and investor reporting logic 30 and the securitization logic 32 or, preferably, uses a separate data base that is synchronized with these data bases. For both whole loans and securities, the position monitor 158 provides daily and month-to-date commitment/trade and delivery/settlement positions. The position monitor 158 also provides cumulative inventory

positions held by the portfolio. The position monitor 158 allows investors to manage inventory from an economic, risk management, and regulatory accounting and taxation perspective. It also allows investors to determine or designate what assets to buy, what assets to sell, and what assets to retain or hold for investment. The portfolio manager 158 provides investors with a clear and concise view of their current net position of inventory.

**[0110]** The out of portfolio (OOP) pooling logic 160 permits the data processing system 12 to be used for pooling loans to create financial instruments in situations where the loans are owned by the entity that owns or operates the data processing system 12 or by an entity other than the entity that owns/operates the data processing system 12. The OOP pooling logic 160 provides the owner of the loans being pooled with the ability to select asset attributes and sub-attributes at a granular level, the ability to select loans to optimize chartered pool statistics, the ability to flexibly map incoming and outgoing cash flows, and the ability to use an on-screen display to manipulate collateral. The out of portfolio pooling processor 160 also has the ability to collateralize asset cashflows as described above in connection with the packeting logic 154.

**[0111]** The whole loan trading logic 162 provides a facility for engaging in whole loan trades to permit the owner or operator of the data processing system 12 to identify and sell loans out of its portfolio to other entities. The whole loan trading logic 162 also provides logic for reporting to the servicer of a sold loan (1) that the loan has been sold and (2) the identity of the new owner of the loan, allowing the servicer to begin reporting payment information to the new owner.

**[0112]** Referring to Fig. 6, the common services logic 34 includes work flow processor 170 which generates notifications about required actions and routes the notifications to users of the data processing system 12 according to

pre-defined processing sequences for request approvals and exception report resolutions. The work flow processor 170 also keeps track of status and actions related to work items.

**[0113]** The report processor 172 generates reports based on users' requests. The report processor 172 allows data to be extracted from the data bases to prepare reports that can be sent out through the user services logic 22. The reports that are returned may be bulk transfers of data. The report processor 172 supports generating the reports described above in connection with the acquisition logic 28, the servicer and investor reporting logic 30, and the securitization logic 32.

**[0114]** The database and access control logic 174 provides database and user security administration and control for the databases in the data storage system 38 and functions available through system 12. The database access and control logic also maintains referential integrity, processes queries and updates, and performs all tasks related to access and control of the databases in the data storage system 38.

**[0115]** The process controller/scheduler 176 triggers execution of processes based on time schedule and/or events received from application components. The process controller/scheduler encapsulates information on processing interdependencies between different components in the data processing system 12.

**[0116]** The audit logging logic 178 logs data that is needed for historical tracking of the activities of the data processing system 12. The purpose of the data logging is primarily to meet audit requirements in connection with the transactions processed by the data processing system 12.

**[0117]** The business rules processor 180 is a rules engine that encapsulates business rules to permit the business rules to be applied to the loan data.

Examples of the business rules applied by the rules processor 180 have been described throughout the discussion of the data processing system 12. A user interface is provided that allows the business rules to be modified and that allows new business rules to be added or obsolete business rules to be deleted. The rules processor 180 maintains the business rules separate from the remainder of the application code that implements other aspects of the data processing system 12. This allows the business rules to be modified/added/deleted without requiring revisions to the application code. The ability to modify or add business rules quickly facilitates the introduction of new types of loan products and investment instruments, because the data processing system 12 may be easily modified to implement any special data processing required for the implementation of the new loan products/investment instruments. Preferably, the rules processor 180 is provided as three separate rules processor, one for each of the acquisition logic 28, the servicer and investor reporting logic 30, and the securitization logic 32, with separate user interfaces for each rules processor.

**[0118]** As previously indicated, service granularity is achieved in part by representing loans as a series of data attributes. The following is an example of a set of attributes that may be used to characterize loans: accounting class code; accounting close effective period; accounting reporting category code; actual UPB at acquisition; adjusted last paid installment date; adjusted unpaid principal balance; ceiling; change frequency; change method; conduit code; custodian code; downward cap; downward cap code; effective date; excess yield; excess yield adjustment; extended term; purchaser loan number; final step change; first PITI (principal, interest, taxes, insurance) due date; fixed interest rate; fixed pass-thru rate; fixed payment amount; floor; frequency of payment change; frequency of rate change; future feature code; index code; index lookback; interest rate; loan guaranty payment date; loan conversion date; loan guaranty date; loan payoff interest calculation code; loan rate effective date;



loan to value ratio; LP control record; lender pass through (LPT) type code; maximum term; months payment control effective; months rate control effective; mortgage margin; mortgage term; net interest adjustment; new payment amount; next control record; next scheduled payment change date; next scheduled rate change date; number of months in effect; other fees collected adjustment; pass-thru rate; payment change amount/percentage; payment change method code; payment control record; payment type code; principal adjustment; processing status code; product code; rate change method code; rate change percent; rate control record; rate conversion status code; rate rounding method; rate type code; reclassification date; remittance day code; required change index; required margin; secured unpaid principal balance; servicing fee; servicing fee adjustment; servicing fee type; servicing remittance option; unpaid principal balance; upward cap; upward cap code. In addition to the above-mentioned attributes, additional attributes may be used in connection with particular types of specialty loan products.

**[0119]** As previously indicated, data granularity is achieved at least in part by decomposing loan assets into a series of cash flows. A cash flow may be any type of payment, whether of principal, interest, or fees. Cash flow may also includes credit-related losses, which manifest themselves from the securities standpoint as negative investor payments (i.e., a reduction to positive cash flows). Possible sources of cash flow may be associated with principal, interest, servicing fees, guarantee fees, mortgage insurance, prepayment penalties, borrower-paid fees, servicer advances, servicer recoveries, loss/default components, and REO activity. For principal, individual cash flows that may be identified include the following: scheduled principal (amount payable based on scheduled amortization), actual principal (what was applied as principal), unscheduled principal (amount from borrower applied in excess of scheduled), advanced (amount not collected from borrower but remitted to investor), shortfall (underpayment from borrower, usually meaning less than full scheduled

amount). For interest, individual cash flows that may be identified include the following: scheduled Interest (amount payable), actual (what was applied), excess (interest collection in excess of amount payable), advanced (not collected from borrower but sent to investor), shortfall (underpayment from servicer), capitalized (negative amortization), other capitalized interest (delinquency), unrecoverable prepayment interest shortfall. For servicing fees, individual cash flows that may be identified include the following: gross servicing fee, core servicing fee (usually relates to tax), excess servicing fee, safe harbor (tax). For guarantee fees, individual cash flows that may be identified include the following cash flows: gross guarantee fee (GF) (total charged to the lender), cash flows for internally tracking costs (e.g., costs associated with credit risk), base GF, GF variance, and other GF adjustments. For mortgage insurance (MI), individual cash flows that may be identified include the following: lender paid MI, borrower paid MI, portion of GF construed to be MI, back-end MI. For prepayment penalties, individual cash flows that may be identified include the following: prepayment penalty, prepayment penalty (borrower-paid), yield maintenance fee (borrower-paid). For borrower-paid fees, individual cash flows that may be identified include the following: borrower-paid fees, late payment fee, conversion/modification fee. For seller advances, individual cash flows that may be identified include the following: advanced principal, advanced interest, advanced guaranty fee, servicing advances (usually relates to defaults, e.g., T&I). For servicer recoveries, individual cash flows that may be identified include the following: recovered principal advances, recovered interest advances, recovered guaranty fee advances, recovered servicing advances. For default activity, cash flows that may be identified include the following: net realized loss (total amount payable to investors less all recoveries), foreclosure expenses, attorney fees, recoup of non-recoverable advances, loss due to modification, loss due to appraisal reduction, loss due to deficiency valuation, non-capitalized deferred interest (e.g. workout), interest paid on advances. For REO activity, cash flows that may be identified include

the following: foreclosure sale proceeds, rental income, insurance proceeds, tax expenses on REO, repair expenses on REO, sale/marketing expenses on REO, REO property maintenance expenses. It may be noted that some of the above cash flows are aggregate cash flows that can be further decomposed. Other cash flow pertinent information that may be tracked includes unpaid principal balance (UPB) (including scheduled UPB and actual UPB), participation percentage (including principal participation percentage, interest participation percentage, and servicing fee participation (basis points)), discount rate (used to calculate yield maintenance or prepayment penalty), appraised balance, foreclosure sale date, and REO sale date.

**[0120]** Many other changes and modifications may be made to the present invention without departing from the spirit thereof. For example, each of the features described above may also be implemented in systems or logic that are configured differently than the data processing system 12 and/or that include different, fewer or more functions than the functions included in the data processing system 12. The scope of these and other changes will become apparent from the appended claims.